Introduction to Cascading Style Sheets

4

LEARNING OBJECTIVES

After completing this chapter, you will be able to understand:

- CSS and how to use it.
- How CSS can transform the look of a page.
- CSS configuration with HTML.
- Advantages of CSS.
- Various selectors like Id, class, etc. and how to use them.
- How to attach an external CSS file.
- Various HTML elements styles.

4.1 | Introduction

In Chapter 3, we have gained a firm grasp of HTML. Now, let us proceed to the next fundamental front-end technology – Cascading Style Sheets (CSS).

CSS falls into the domain of style sheet languages. It is primarily used for defining the presentation of web pages. While HTML defines the structure of the complete web page, CSS adds some style to it. Without CSS, websites look ugly and bland. Investing in CSS is a good way to add appeal to the website and increase web traffic.

We can use single CSS file with elements such as the layout, fonts, color, and other related elements of a webpage which can be written once and used many times. By using a single CSS file, the layout, fonts, color, and other related elements of a webpage can be sorted out all at once. CSS files are saved with an extension of ".CSS".

4.1.1 History

On October 10, 1994, Håkon Wium Lie proposed CSS. During that time, Lie used to work at CERN with Tim Berners-Lee. Many other style sheet languages were also proposed at that time and the World Wide Web Consortium held discussions on the matter. After some time, CSS1 was released in 1996. This release had substantial input from Bert Bos who was the co-author and is known as the CSS' co-creator.

4.2 | Overview of CSS

As explained, CSS is a language which is used to define how users can view documents and understand how they are laid out or styled. This document is generally a text file which uses a markup language. Most of the time, this markup language is HTML. However, you may also have to use CSS with XML, Scalable Vector Graphics (SVG), or other markup languages.

By "presenting" this document we mean that it must be converted into a form which is usable by the target audience. Chrome, Firefox, and other web browsers are created with the functionality to present these documents on a computer screen, printer, or a project.

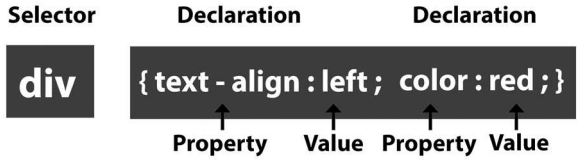
4.3 | Relationship Between HTML and CSS

The CSS rules are used by the web browsers with a document to change how it appears to the user. Such rules are designed from the following stages:

1. **Properties:** These properties are configured with values to modify how the content of HTML should be displayed to the user. For example, you can set the width of a button on the page or change its color by updating the appropriate property.

2. **Selector:** It chooses the elements by their ids, names, types, attributes, etc., on which the desired style is going to be applied. For example, you can use it to ensure that all the paragraphs have green text color.

CSS rules in the style sheet ultimately form the look of the page.



4.4 | How Does CSS Work?

After a document is displayed by the browser, it has to use the content of the document and incorporate it with the style sheet information. The browser performs this processing using the following strategy:

1. The browser changes CSS and HTML into the Document Object Model (DOM). This model is used by computer's memory for representation of the document.
2. The DOM's contents are displayed by the browser.

The DOM model consists of a structure which resembles a tree data structure. All the attributes, elements, and text of the markup language are added as the DOM node in this structure. Each node has a specified relationship with the other. This means that a node can have a parent, child, and siblings.

If you can understand the DOM model, it can assist you to design, maintain, and debug the CSS as the DOM describes how the content of the document and the CSS have to be linked.

4.4.1 Advantages of Using CSS

1. CSS helps in saving time. It is only needed to be defined once, after which it can be used again and again to style all HTML elements for several HTML pages.
2. It helps to load web pages quicker. When CSS is used, there is no need for HTML tag attributes. Developers have to use the CSS rule for a tag one time, after which it is automatically implemented for all of the tag's occurrences. Hence, it decreases code and ultimately helps to increase the speed of downloading.
3. For a global change, you can use the CSS to just change the code in a single place which can apply the changes to all the relevant webpages. Hence, it is extremely easy to maintain CSS.
4. In comparison to HTML, CSS has a large variety of attributes which helps to add several new features and functionalities that were otherwise not possible with the use of HTML attributes.
5. CSS can help the content with optimization which means that it can display appropriately in multiple devices. CSS can utilize the same HTML which can show multiple website versions for smartphones, tablets, printers, and PDAs.
6. Styling webpages with HTML attributes is considered obsolete today as there are several disadvantages. First is the readability of the code where developers cannot easily figure out the styling of the element; adding more styling properties will become cumbersome. Second, it adds to the page load time since in case of separate CSS files, the browser can download HTML and CSS files simultaneously to save loading time. Hence, using CSS is a good design practice.

4.4.2 HTML and Styling

HTML also has several options to add "styling" on the web page, so why should developers move towards CSS? Well, in four words we can explain the practice as per our personal experience: It is a nightmare! Adding style to large websites using HTML is extremely difficult because web designers have to make modifications to each page. To tackle this issue, the World Wide Web Consortium (W3C) came up with CSS. Unlike other formatting tools, CSS has completely eliminated the need for HTML as an alternate webpage styling option.

4.5 | Syntax

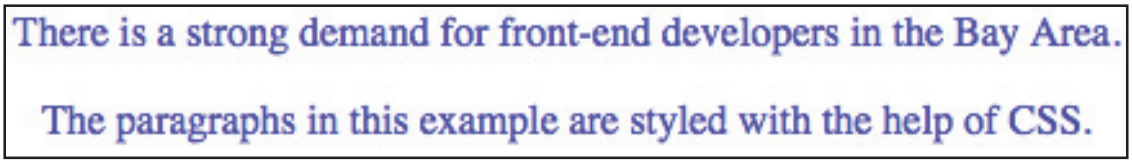
With enough understanding about the necessity of CSS, let us now move to the more technical aspects of CSS.

In CSS, we have two components: a selector and a declaration block. A selector refers to any HTML element which requires styling. However, it is not necessary for a selector to be the name of an element. We can also use the attribute *id*, or *class*, as selectors.

By a declaration block, we mean the set of CSS properties with their values. These properties are separated by a colon. A declaration block may contain a single property or multiple properties. The block is terminated with a curly bracket, while the entailed properties are ended with curly braces.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    p {
        color: blue;
        text-align: center;
    }
    </style>
</head>
<body>
    <p>There is a strong demand for front-end developers in the Bay Area.</p>
    <p>The paragraphs in this example are styled with the help of CSS.</p>
</body>
</html>
```

This code shows the following result in a browser window.

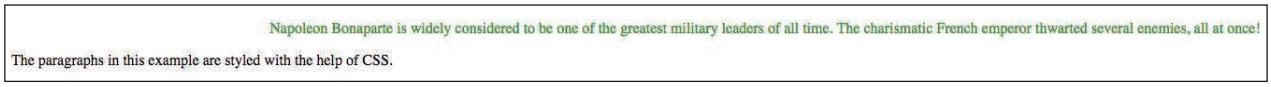


4.5.1 The "*id*" Selector

CSS can use selectors to select a specific element to apply the desired style. For example, for an HTML element with an "*id*" attribute set to "myID", CSS can use id selector such as #myID { color: red; } to apply the defined styling to that HTML element. In this case, the HTML element with id "myID" will get red color as defined in the CSS selector. Note that the attribute in HTML must be unique, that is, no other element should have the same id. Moreover, an id name must never begin with a digit. To use an id selector, you have to type "#" (hash character), after which you can define an element's id. For example,

```
<!DOCTYPE html>
<html>
<head>
    <style>
    #firstpara {
        text-align: right;
        color: green;
    }
    </style>
</head>
<body>
    <p id="firstpara">Napoleon Bonaparte is widely considered to be one of the greatest
        military leaders of all time. The charismatic French emperor thwarted several
        enemies, all at once!</p>
    <p>The paragraphs in this example are styled with the help of CSS.</p>
</body>
</html>
```

This code shows the following result in a browser window.



QUICK CHALLENGE

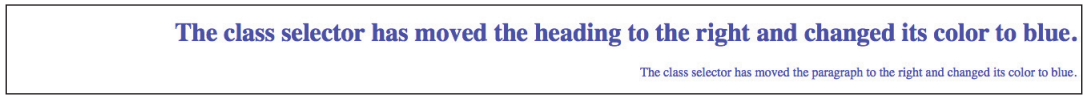
Add multiple elements with same id and see what happens. For example, create three tags and give them same id like id= "myDiv". Then use selector to style those div elements. Note down the outcome and write lesson learned in a few sentences.

4.5.2 The "*class*" Selector

As its name suggests, this type of selector chooses elements that carry a defined class attribute. The class name is preceded by a ".". For example,

```
<!DOCTYPE html>
<html>
<head>
    <style>
    .styling {
        text-align: right;
        color: blue;
    }
    </style>
</head>
<body>
    <h1 class="styling">The class selector has moved the heading to the right and changed
        its color to blue.</h1>
    <p class="styling">The class selector has moved the paragraph to the right and
changed its color to blue.</p>
</body>
</html>
```

This code shows the following result in a browser window.



There is also the flexibility to only apply changes in specific HTML elements of the class. This can be done by using the name of the element. Continuing our above example we have,

```
<!DOCTYPE html>
<html>
<head>
    <style>
    p.styling {
        text-align: right;
        color: blue;
    }
    </style>
</head>
<body>
    <h1 class="styling">No changes were applied to the heading.</h1>
    <p class="styling">The class selector has moved the paragraph to the right and
changed its color to blue.</p>
</body>
</html>
```

The above code shows the following result in a browser window.

4.5.3 Groups of Selectors

Consider the following CSS code. As you can observe, all of these elements like h1, h2 and p have the same functionality, but their declaration blocks are repeated. Hence, it seems quite an un-optimized way of coding.

```
h1 {
    text-align: left;
    color: blue;
}
h2 {
    text-align: left;
    color: blue;
}
p {
    text-align: left;
    color: blue;
}
```

However, in CSS, it is possible to group selectors and incorporate similar functionality where a comma demarcates them. Therefore, we can optimize the above code and reduce our lines of CSS considerably by writing the following code:

```
<!DOCTYPE html>
<html>
<head>
    <style>
    h1, h2, p {
        text-align: left;
        color: blue;
    }
    </style>
</head>
<body>
    <h1>First Heading</h1>
    <h2>Second Heading</h2>
    <p>This paragraph is blue.</p>
</body>
</html>
```

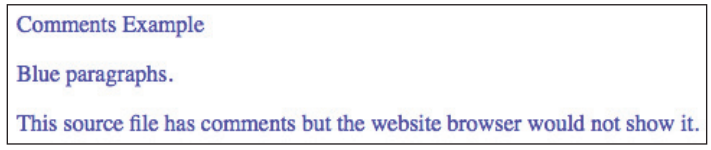
This code shows the following result in a browser window.

4.5.4 Comments

Web designers and developers use comments in CSS to write useful details for lines of code. These notes come in handy when you want to change the code later on. Website browsers do not display comments. Comments begin with /* and end with */. For example,

```
<head>
    <style>
    p {
        color: blue;
        /* We have applied this CSS setting because of client requirements */
        text-align: left;
    }
    /* Comments
    can extend
    multiple lines */
    </style>
</head>
<body>
    <p>Comments Example</p>
    <p>Blue paragraphs.</p>
    <p>This source file has comments but the website browser would not
        show it.</p>
</body>
</html>
```

This code shows the following result in a browser window.



Flash Quiz

Will a page throw error if CSS code is not in proper format?

4.6 | Different Methods to Integrate CSS with HTML

In this section we will discuss the various methods using which a style sheet can be inserted.

4.6.1 External Style Sheet

External style sheets are separate files that carry the **.css** extension. They simplify the presentation of webpages because they store all modifications in a single file. Style sheets are referenced in all the pages of the websites in the *<link>* element, which itself must be entailed in the *<head>* element. To make an external style sheet, open a file in Notepad and save it with "*cssexample.css*". Add the following instructions in the file:

```
p {
    text-align: right;
    color: blue;
}
```

Now for HTML, write this code.

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="cssexample.css">
</head>
<body>
    <h1>External Style Sheet Example</h1>
        <p>This paragraph is styled with CSS.</p>
</body>
</html>
```

This code shows the following result in a browser window.

Note that the CSS files do not use any of the HTML's tags.

4.6.2 Internal Style Sheet

Internal style sheets are used with the HTML element *<style>* inside the *<head>* element. Remember, the *<head>* element is used for the metadata of HTML. So, why do we use internal style sheets? The purpose behind the use of internal style sheet is to add a separate or unique style to a page. For instance,

```
<!DOCTYPE html>
<html>
<head>
    <style>
    body {
        background-color: green;
    }

    h1 {
        color: brown;
        margin-right: 100px;
    }
    </style>
</head>
<body>
    <h1>This is a brown heading</h1>
        <p>Green background is used in the background.</p>
</body>
</html>
```

The above code shows the following result in a browser window, where the background is shown in green and the main heading text is in red.

4.6.3 Inline Style

The third method to use CSS is inline styling. In this case, styling is applied on single elements for unique functionality. They are incorporated in HTML elements by adding the style attribute in the desired element. This style attribute can now make use of any CSS property. For instance,

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="color: green; margin-right: 30px;">This is a green heading with inline
styles.</h1>
    <p>No CSS is used here.</p>
</body>
</html>
```

The above code shows the following result in a browser window.

4.6.3.1 Use of Multiple Style Sheets

Sometimes, different style sheets refer to the same HTML element. A style sheet may force a heading to go blue while another one to turn green. So how would the browser know which one to choose? In such cases, the last read style sheet is taken into account for the changes.

4.7 | Colors

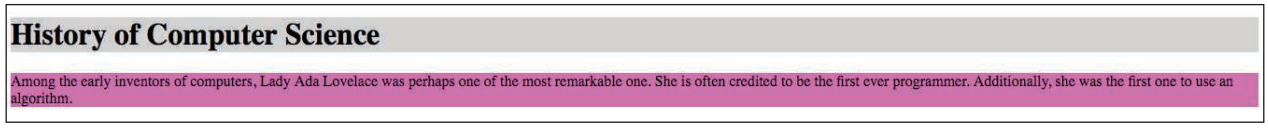
In CSS, colors are defined in multiple ways. You can use common color names, or you can add RGBA, HSLA, HSL, HEX, and RGB color values.

4.7.1 Color Names

All elements in HTML can have their own background color by using the "*background-color*" property. For instance,

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="background-color: LightGray;">History of Computer
        Science</h1>
    <p style="background-color: Violet;">Among the early inventors of computers, Lady Ada
       Lovelace was perhaps one of the most remarkable one. She is often credited to be the
       first ever programmer. Additionally, she was the first one to use an algorithm.</p>
</body>
</html>
```

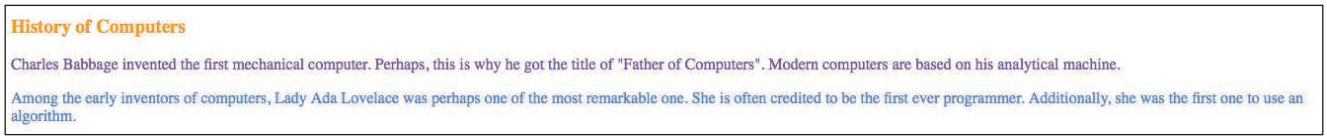
This code shows the following result in a browser window.



Similarly, the text of the HTML elements can also be updated by using the "*color*" property.

```
<!DOCTYPE html>
<html>
<body>
    <h3 style="color: orange;">History of Computers</h3>
    <p style="color: slateblue;">Charles Babbage invented the first mechanical computer.
       Perhaps, this is why he got the title of "Father of Computers". Modern computers
       are based on his analytical machine.</p>
    <p style="color: dodgerblue;">Among the early inventors of computers, Lady Ada
       Lovelace was perhaps one of the most remarkable one. She is often credited
       to be the first ever programmer. Additionally, she was the first one to use an
       algorithm.</p>
</body>
</html>
```

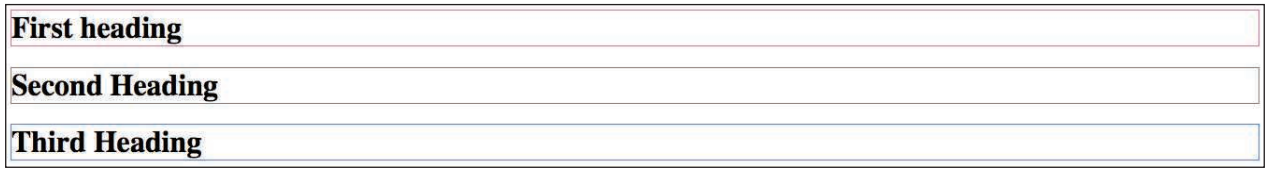
This code shows the following result in a browser window where h3 text is shown in orange, first p element text is shown in slate blue and second p element text is shown in Dodger blue.



Similarly, the border color of elements is also modifiable.

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="border: 1px solid Violet;">First heading</h1>
    <h1 style="border: 1px solid Gray;">Second Heading</h1>
    <h1 style="border: 1px solid DodgerBlue;">Third Heading</h1>
</body>
</html>
```

This code shows the following result in a browser window where first h1 element has violet border, second h1 has gray border, and third h1 has Dodger blue border.



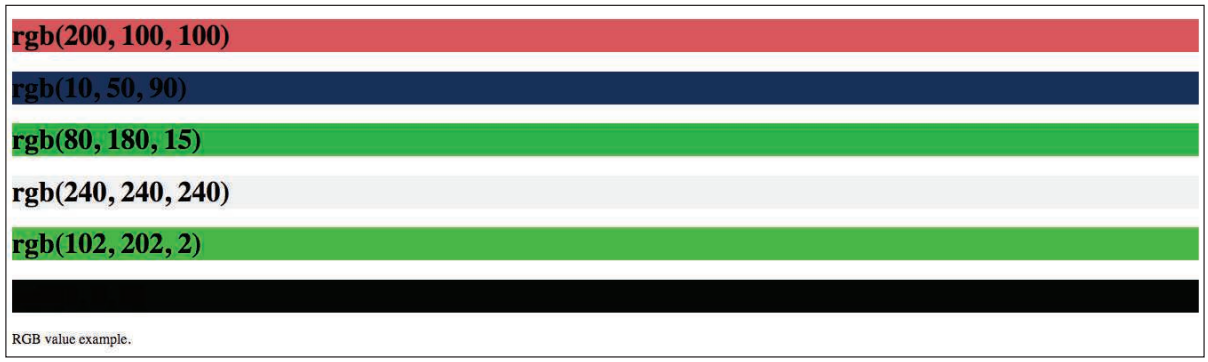
4.7.2 Color Values

As mentioned above, colors can also be defined using the HEX, HSL, HSLA, RGBA, and RGB values.

To use the RGB values, three separates values are required for red, green, and blue colors. These values can store a digit in the range of 0–255, where each digit represents the intensity of the color. For instance, a RGB value of (0, 255, 0) refers to the use of green color because its intensity is set to the max value for green color while red and blue are specified as having 0 values.

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="background-color: rgb(200, 100, 100);">rgb(200, 100,100)</h1>
    <h1 style="background-color: rgb(10, 50, 90);">rgb(10, 50, 90)</h1>
    <h1 style="background-color: rgb(80, 180, 15);">rgb(80, 180, 15)</h1>
    <h1 style="background-color: rgb(240, 240, 240);">rgb(240, 240, 240)</h1>
    <h1 style="background-color: rgb(102, 202, 2);">rgb(102, 202, 2)</h1>
    <h1 style="background-color: rgb(1, 2, 3);">rgb(1, 2, 3)</h1>
    <p>RGB value example.</p>
</body>
</html>
```

This code shows the following result in a browser window where first h1 element's background is in maroon, second h1 element's background is in navy blue, third h1 element's background is in dark green, fourth h1 element's background is in light grey, fifth h1 element's background is in green, and sixth h1 element's background is in black.



Likewise, you can also use the hexadecimal format for specifying colors. The format is #rrggbb which stores hexadecimal values starting from 00 to ff. For instance, #00ff00 refers to green because green's intensity is set to its maximum value in the hexadecimal notation.

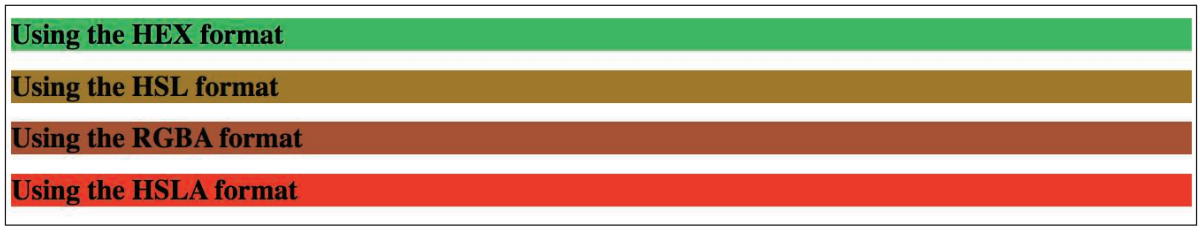
Similarly, HSL can also be used for defining colors. HSL comprises hue, saturation, and lightness. Hue is represented by a degree which falls into the range of 0 to 360 on a color wheel. Saturation and lightness are both represented by percentage values where the former applies a shade of gray while the latter applies darkness and whiteness on the colors.

We can also use the RGBA format for specifying colors. RGBA is same as RGB, except for the addition of A which refers to the alpha channel, which represents the opacity of the color.

Lastly, we have the HSLA color values which is similar to the HSL color values except for the addition of A which here also refers to the alpha channel; it represents opacity of the color. Let us see the use of all color values in the following example,

```
<!DOCTYPE html>
<html>
<body>
    <h1 style="background-color: #2bc262;">Using the HEX format</h1>
    <h1 style="background-color: hsl(40, 50%, 40%);">Using the HSL format</h1>
    <h1 style="background-color: rgba(155, 88, 61, 50);">Using the RGBA format</h1>
    <h1 style="background-color: hsla(7, 66%, 55%, 20);">Using the HSLA format</h1>
</body>
</html>
```

This code shows the following result in a browser window where first h1 element's background is in green, second h1 element's background is in light brown, third h1 element's background is in dark brown, and fourth h1 element's background is shown in red.



Flash Quiz

Are there any advantages of using color values over color names?

4.8 | Backgrounds in CSS

CSS offers several properties to edit the background of HTML elements. To add color to the background of an HTML element, you can use the background-color property as follows:

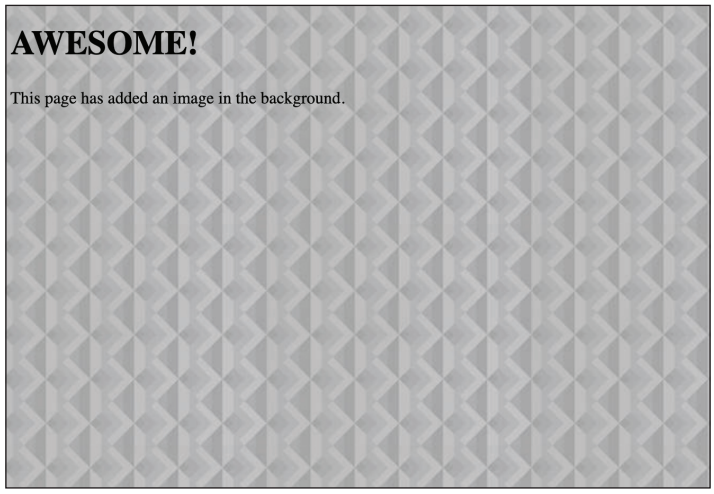
```
<!DOCTYPE html>
<html>
<head>
    <style>
    body {
        background-color: orange;
    }
    </style>
</head>
<body>
    <h1>Background Color Property</h1>
    <p>Orange background</p>
</body>
</html>
```

This code shows the following result in a browser window where entire page's background is shown in orange color.

Similarly, you can also add an image in the background of your element. The following code shows the repetition of image below which completes the pattern for full page.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    body {
     background-image: url("https://www.everythingtech.co/wp-content/uploads/2019/11
/verticalpattern250img.jpg");
    }
    </style>
</head>
<body>
    <h1>AWESOME!</h1>
    <p>This page has added an image in the background.</p>
</body>
</html>
```

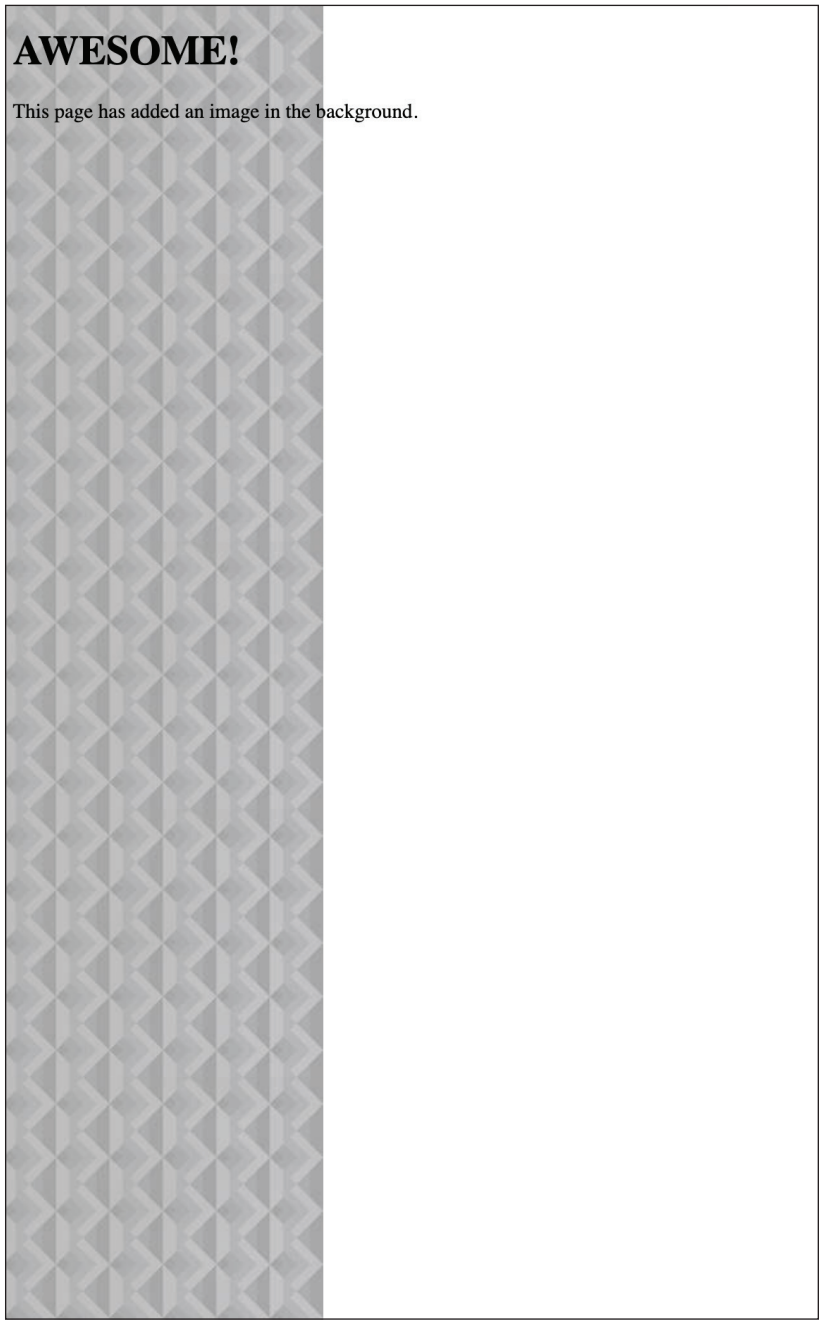
This code shows the following result in a browser window.



Some images look better if they are repeated either vertically or horizontally. To perform the repetition of an image vertically, use *background-repeat: repeat-y property* and for horizontal display, use *background-repeat: repeat-x format*. The following code shows the repetition of image below vertically which completes the pattern:

```
<!DOCTYPE html>
<html>
<head>
   <style>
   body {
      background-image:
         url("https://www.everythingtech.co/wp-content/uploads/2019/11
/verticalpattern250img.jpg");
      background-repeat: repeat-y;
   }
   </style>
</head>
<body>
   <h1>AWESOME!</h1>
   <p>This page has added an image in the background.</p>
</body>
</html>
```

This code shows the following result in a browser window.



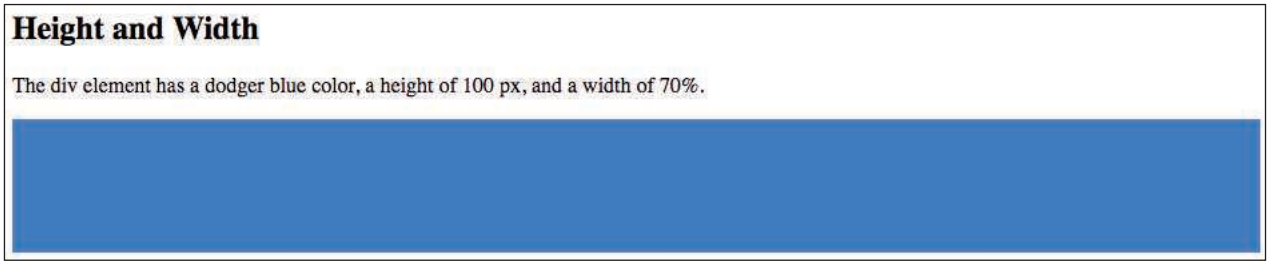
What if you do not want to repeat an image? Well, then you just need to add the background-repeat property. Sometimes, a website requires for the image to be present at all times, even while scrolling. To do this, use the *background-attachment:fixed* property. Do remember one thing: Place the image in a way that it does not affect the readability of your text.

4.9 | Setting up Height and Width of an Element

CSS can help to specify the height/width of an HTML element by using the properties "*height*" and "*width*". They can be either defined specifically in *cm*, *px*, *%* or they can be simply specified as "*auto*". The browser would automatically generate the height and width.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        height: 100px;
        width: 70%;
        background-color: dodgerblue;
    }
    </style>
</head>
<body>
    <h2>Height and Width</h2>
    <p>The div element has a dodger blue color, a height of 100 px, and a width of 70%.</p>
    <div></div>
</body>
</html>
```

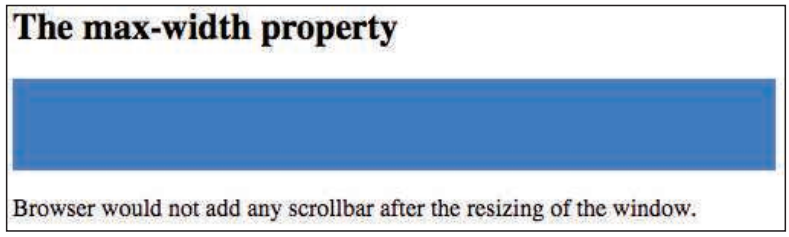
This code shows the following result in a browser window where the p element background is shown in Dodger blue color.



When the width of the browser is lower than the element's width, then it places a horizontal scroll bar in the webpage. To address this concern, the *max-width* property is used to specify the element's maximum width.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        max-width: 500px;
        height: 60px;
        background-color: dodgerblue;
    }
    </style>
</head>
<body>
    <h2>The max-width property</h2>
    <div></div>
    <p>Browser would not add any scrollbar after the resizing of the window.</p>
</body>
</html>
```

This code shows the following result in a browser window.



The browser no longer adds a scrollbar as the element repositions itself automatically.

4.10 | Box Model

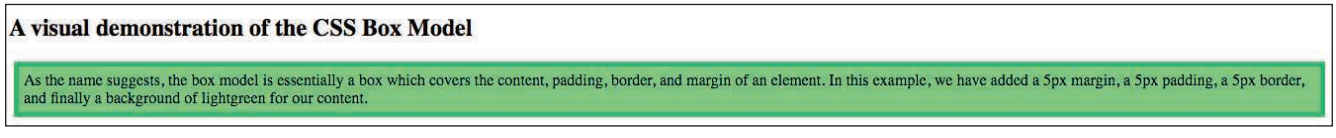
A certain type of model is often discussed among web designers who work with CSS. This model, known as the *box model*, can reshape the layout and design of a webpage. It is called "box" because it is really a box which wraps around every HTML element. It comprises the following:

1. **Margins:** A transparent property which clears the portion encircling the border.
2. **Borders:** A non-transparent property which encircles both the content and the padding.
3. **Padding:** A transparent property which clears the portion encircling the content.
4. **Content:** Information for readers (i.e., text and images).

Run the below code to view a practical example of the box model.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        background-color: lightgreen;
        border: 5px solid mediumseagreen;
        padding: 5px;
        margin: 5px;
    }
    </style>
</head>
<body>
    <h2>A visual demonstration of the CSS Box Model</h2>
    <div>As the name suggests, the box model is essentially a box
        which covers the content, padding, border, and margin of an element.
        In this example, we have added a 5px margin, a 5px padding, a 5px
        border, and finally a background of lightgreen for our content.</div>
</body>
</html>
```

This code shows the following result in a browser window where the div element background is in lightgreen and border is in medium sea green.



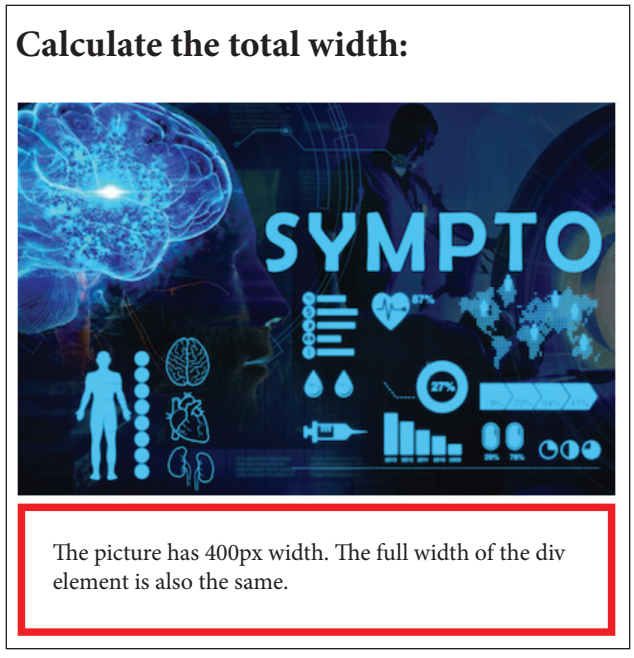
Understanding the box model is important for setting up the width and height of HTML elements in the web browsers. For instance, suppose we have to specify a width of 400px for an element. To specify it, we would have to stick to the following formula:

Full Element Width = *Width* + *Padding* (*left and right*) + *Border* (*left and right*) + *Margin* (*left and right*)

Similarly, for the complete element height, use the following formula:
Full Element Height = *Height* + *Padding* (*top and bottom*) + *Border* (*top and bottom*) + *Margin* (*top and bottom*)

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div {
        width: 350px;
        padding: 20px;
        border: 5px solid red;
        margin: 0;
    }
    </style>
</head>
<body>
    <h2>Calculate the total width:</h2>
    <img
        src="https://www.everythingtech.co/wp-content/uploads/2019/10/symptosmall.jpg"
        width="400" height="263" alt="Sympto Open Source Project">
    <div>The picture has 400px width. The full width of the div
        element is also the same.</div>
</body>
</html>
```

This code shows the following result in a browser window where the div element border is in red color.



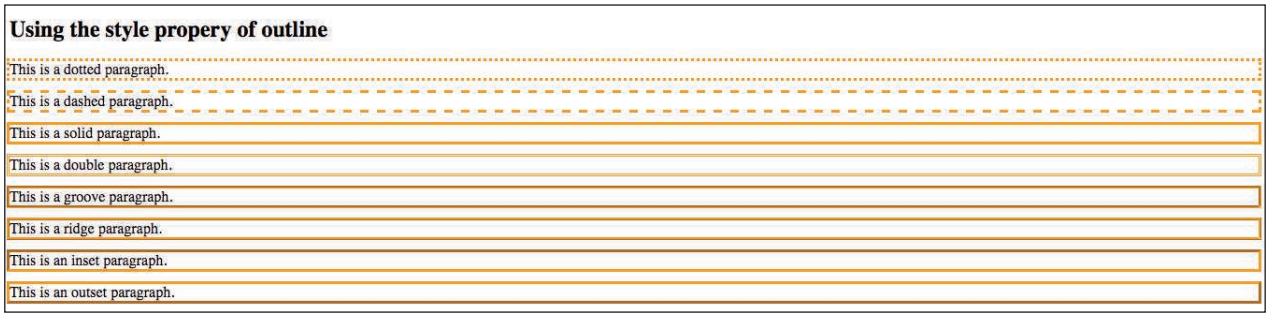
QUICK CHALLENGE Implement multiple box models on a page and view the result in various different browsers with different resolutions.

4.11 | CSS Outline

In CSS, an outline refers to a line which is created on the sides of the element on the border's outside so the element can become more "prominent". The outline style is defined by the *outline-style* property. It can have several values like: *hidden*, *outset*, *inset*, *none*, *ridge*, *double*, *solid*, *groove*, *ridge*, *dotted*, and *dash*.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    p {
        outline-color: orange;
    }

    p.a {
        outline-style: dotted;
    }

    p.b {
        outline-style: dashed;
    }

    p.c {
        outline-style: solid;
    }

    p.d {
        outline-style: double;
    }

    p.e {
        outline-style: groove;
    }

    p.f {
        outline-style: ridge;
    }

    p.g {
        outline-style: inset;
    }

    p.h {
        outline-style: outset;
    }
    </style>
</head>
<body>
    <h2>Using the style property of outline</h2>
    <p class="a">This is a dotted paragraph.</p>
    <p class="b">This is a dashed paragraph.</p>
    <p class="c">This is a solid paragraph.</p>
    <p class="d">This is a double paragraph.</p>
    <p class="e">This is a groove paragraph.</p>
    <p class="f">This is a ridge paragraph.</p>
    <p class="g">This is an inset paragraph.</p>
    <p class="h">This is an outset paragraph.</p>
</body>
</html>
```

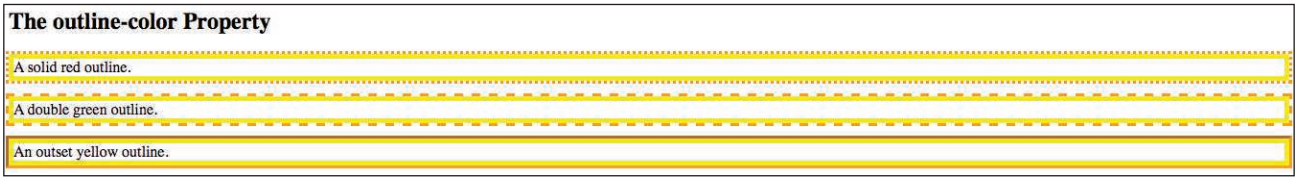
This code shows the following result in a browser window:



To specify a color for the outline, the *outline-color* property can be used. These colors can be defined in the following formats: *name*, *Hex* and *RGB* values, and invert–applying a color inversion.

```
<!DOCTYPE html>
<html>
<head>
   <style>
   p.para1 {
      border: 5px solid yellow;
      outline-style: dotted;
      outline-color: orange;
   }

   p.para2 {
      border: 5px solid yellow;
      outline-style: dashed;
      outline-color: orange;
   }

   p.para3 {
      border: 5px solid yellow;
      outline-style: inset;
      outline-color: orange;
   }
   </style>
</head>
<body>
   <h2>The outline-color Property</h2>
   <p class="para1">A solid red outline.</p>
   <p class="para2">A double green outline.</p>
   <p class="para3">An outset yellow outline.</p>
</body>
</html>
```

This code shows the following result in a browser window where all the p elements' borders are in yellow and outlines are in orange color.



To define the outline's width, the outline-width property is used. It can accept the following values: a specific input (*pt*, *px*, *cm*, *em*), thick (usually 5px), medium (usually 3px), and thin (usually 1px).

```
<!DOCTYPE html>
<html>
<head>
    <style>
    p.para1 {
        border: 1px solid green;
        outline-style: solid;
        outline-color: orange;
        outline-width: thin;
    }

    p.para2 {
        border: 1px solid green;
        outline-style: solid;
        outline-color: orange;
        outline-width: medium;
    }

    p.para3 {
        border: 1px solid green;
        outline-style: solid;
        outline-color: orange;
        outline-width: thick;
    }

    p.para4 {
        border: 1px solid green;
        outline-style: solid;
        outline-color: orange;
        outline-width: 4px;
    }
    </style>
</head>
<body>
    <h2>The outline-width Property</h2>
    <p class="para1">This is an example of a thin outline.</p>
    <p class="para2">This is an example of a medium outline..</p>
    <p class="para3">This is an example of a thick outline.</p>
    <p class="para4">This is an example of a 6px thick outline.</p>
</body>
</html>
```

This code shows the following result in a browser window.

For convenience, you can also use the "*outline*" property as an effective shortcut to use these properties: *outline-width*, *outline-style*, and *outline-color*.

4.12 | Text in CSS

With CSS, you can apply a number of different properties on your website's text. For instance, to add a color to your text, you simply use the *color* property via its name or an RGB and HEX value. Likewise, to add the alignment of text such as how it needs to appear on the page, the *text-align* property is used. As an example of using both the *color* and *text-align* properties, consider the following instance:

```
<!DOCTYPE html>
<html>
<head>
    <style>
    h1 {
        text-align: center;
        color: green;
    }

    h2 {
        text-align: left;
        color: green;
    }

    h3 {
        text-align: right;
        color: green;
    }
    </style>
</head>
<body>
    <h1>First Heading Is On The Center</h1>
    <h2>Second Heading Is On The Left</h2>
    <h3>Third heading Is On The Right</h3>
</body>
</html>
```

This code shows the following result in a browser window where all elements text is shown in green color.

For addition or removal of "decoration" on the text, the *text-decoration* property is used. It is sometimes used to eliminate the default underline from the hyperlinks. To check all the possible *text-decoration* values, run the following code:

```
<!DOCTYPE html>
<html>
<head>
    <style>
    h1 {
     text-decoration: overline;
    }
    h2 {
     text-decoration: line-through;
    }
    h3 {
     text-decoration: underline;
    }
    a {
     text-decoration: none;
    }
    </style>
</head>
<body>
    <h1>This is text decoration: overline</h1>
    <h2>This is text decoration: line-through</h2>
    <h3>This is text decoration: underline</h3>
    <p>The underline is removed from the link. <a href="https://google.com">Google</a></p>
</body>
</html>
```

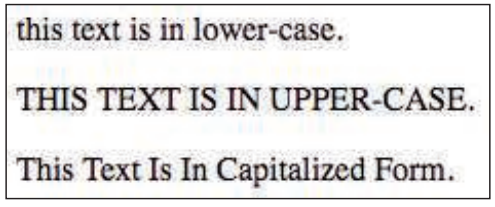
This code shows the following result in a browser window.

It is suggested to avoid using "*text-decoration: underline*" because it can be confusing for the visitor who may click it thinking of it as a hyperlink.

You can also apply "transformation" on the text using the *text-transform* property. Transformation means to apply the lower-case, upper-case, or capitalized format on the text.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    p.lc {
        text-transform: lowercase;
    }

    p.uc {
        text-transform: uppercase;
    }

    p.cp {
        text-transform: capitalize;
    }
    }
    </style>
</head>
<body>
    <p class="lc">This text is in lower-case.</p>
    <p class="uc">This text is in upper-case.</p>
    <p class="cp">This text is in capitalized form.</p>
</body>
</html>
```

This code shows the following result in a browser window.



To apply indentation on the text, the *text-indent* property is utilized. It indents the beginning line of a text. Similarly, to add space between characters, letter-spacing is used.

```
<!DOCTYPE html>
<html>
<head>
   <style>
   p {
      text-indent: 120px;
      letter-spacing: 5px;
   }
   </style>
</head>
<body>
   <p>Many people believe that Leonardo Di Caprio deserved the Oscar
      more for Shutter Island and The Wolf of Wall Street instead of his win
      for the Revenant. Irrespective of this sentiment, he is indeed one of
      the best actors of this generation. His movies with Martin Scorsese
      are a treat to watch.</p>
</body>
</html>
```

This code shows the following result in a browser window.

Many people believe that Leonardo Di Caprio deserved the Oscar more for Shutter Island and The Wolf of Wall Street instead of his win for the Revenant. Irrespective of this sentiment, he is indeed one of the best actors of this generation. His movies with Martin Scorcese are a treat to watch.

4.13 | Fonts

A plethora of font properties exist in CSS to specify the size, boldness, and font family of text. CSS groups font families into two types: generic and font. These properties are defined by using the *font-family* property. Usually, web designers add several fonts as backups because sometimes a browser may not support a specific font. When a font family has a name, which extends to more than a single word, then it is encompassed by quotation marks.

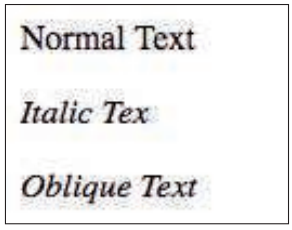
```
<!DOCTYPE html>
<html>
<head>
   <style>
   p.s {
      font-family: "Times New Roman", Times, serif;
   }
   p.ss {
      font-family: Arial, Helvetica, sans-serif;
   }
   </style>
</head>
<body>
   <h1>Font-family Examples</h1>
   <p class="s">This text is using the Times New Roman font.</p>
   <p class="ss">This text is using the Arial font.</p>
</body>
</html>
```

The above code shows the following result in a browser window.

To define the style of the text, font-style is used. It is composed of three values: *italic*, *normal*, and *oblique*.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    p.n {
        font-style: normal;
    }

    p.i {
        font-style: italic;
    }

    p.o {
        font-style: oblique;
    }
    </style>
</head>
<body>
    <p class="n">Normal Text</p>
    <p class="i">Italic Tex</p>
    <p class="o">Oblique Text</p>
</body>
</html>
```

The above code shows the following result in a browser window.



4.14 | Links in CSS

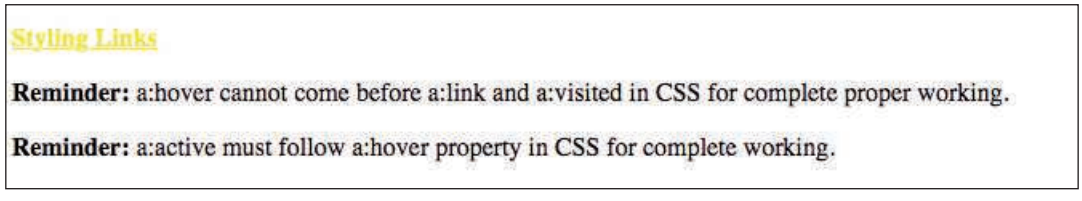
So far, we have used links in many examples. However, they are bland and unappealing. Let us make them stylish and more "functional". All the CSS properties like *background*, *color*, *font-family* can be applied to the links. Additionally, you can add four states in links namely: *a:link*, *a:visited*, *a:hover*, and *a:active*. Let us see these states in detail:

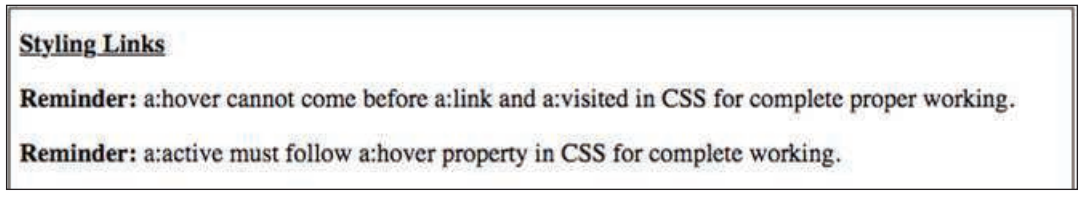
a:link: This state is for a normal unvisited link
a:visited: This state is for a link which is visited by a user
a:hover: This state is for a link on which user has moved over mouse
a:active: This state is when user clicks on the link

Check the following example for their effect on user interaction. In this demonstration, the link initially has a yellow color. Then when it is clicked and visited, its color changes to blue. When the mouse is hovered around it, it changes into black and whenever it is clicked, the color changes to red.

```
<!DOCTYPE html>
<html>
<head>
   <style>
   a:link {
      color: yellow;
   }

   a:visited {
      color: blue;
   }

   a:hover {
      color: black;
   }

   a:active {
      color: red;
   }
   </style>
</head>
<body>
   <p>
      <b><a href="https://google.com" target="_blank">Styling
            Links</a></b>
   </p>
   <p>
      <b>Reminder:</b> a:hover cannot come before a:link and a:visited in
      CSS for complete proper working.
   </p>
   <p>
      <b>Reminder:</b> a:active must follow a:hover property in CSS for
      complete working.
   </p>
</body>
</html>
```

The above code shows the following result in a browser window.





By using all the previous properties of CSS, let's create a link button.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    a:link, a:visited {
        background-color: dodgerblue;
        color: hotpink;
        padding: 10px 20px;
        text-align: left;
        display: inline-block;
    }

    a:hover, a:active {
        background-color: GREEN;
    }
    </style>
</head>
<body>
    <a href="google.com" target="_blank">This is a Link Button.</a>
</body>
</html>
```

This code shows the following result in a browser window where a:hover and a:active states' background color is in green and a:link and a:visited states' background color is in Dodger blue.

Flash Quiz

What is the advantage of using a:visited?

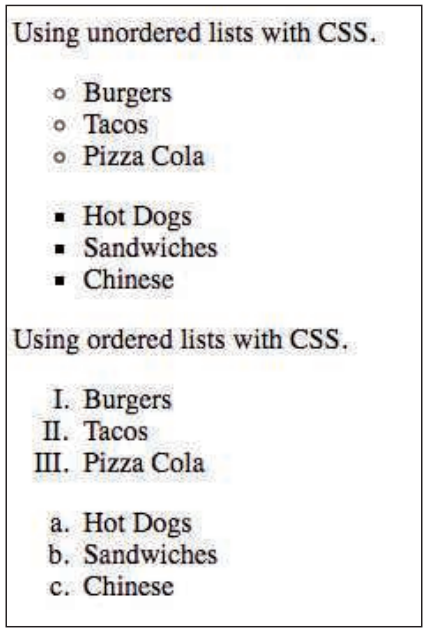
4.15 | Lists in CSS

CSS allows web designers to specify different list item markers for both unordered and ordered lists. Additionally, it facilitates developers to use an image for the marker and incorporate a background color for the list.

By using the *list-style-type* property, the marker of list items can be modified. For instance, consider the following code:

```
<!DOCTYPE html>
<html>
<head>
    <style>
    ul.b1 {
        list-style-type: circle;
    }

    ul.b2 {
        list-style-type: square;
    }

    ol.b3 {
        list-style-type: upper-roman;
    }

    ol.b4 {
        list-style-type: lower-alpha;
    }
    </style>
</head>
<body>
    <p>Using unordered lists with CSS.</p>
    <ul class="b1">
        <li>Burgers</li>
        <li>Tacos</li>
        <li>Pizza Cola</li>
    </ul>
    <ul class="b2">
        <li>Hot Dogs</li>
        <li>Sandwiches</li>
        <li>Chinese</li>
    </ul>
    <p>Using ordered lists with CSS.</p>
    <ol class="b3">
        <li>Burgers</li>
        <li>Tacos</li>
        <li>Pizza Cola</li>
    </ol>
    <ol class="b4">
        <li>Hot Dogs</li>
        <li>Sandwiches</li>
        <li>Chinese</li>
    </ol>
</body>
</html>
```

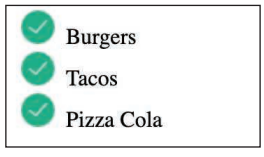
This code shows the following result in a browser window.



For adding an image for the list marker, use the list-style-image property as follows:

```
<!DOCTYPE html>
<html>
<head>
    <style>
    ul {
        list-style-image:
            url('https://www.everythingtech.co/wp-content/uploads/2019/11/tick24.png');
    }
    </style>
</head>
<body>
    <ul>
        <li>Burgers</li>
        <li>Tacos</li>
        <li>Pizza Cola</li>
    </ul>
</body>
</html>
```

This code shows the following result in a browser window.

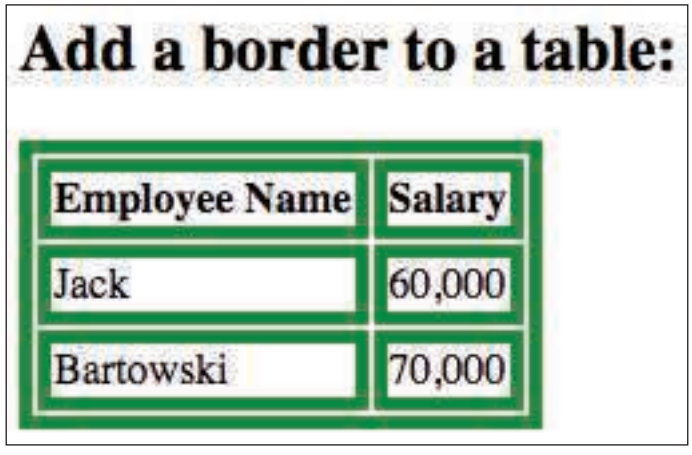


4.16 | Tables in CSS

CSS can help to define different segments of HTML tables. For instance, you can use the *border* property to specify a green border for your table.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    table, th, td {
        border: 5px solid green;
    }
    </style>
</head>
<body>
    <h2>Add a border to a table:</h2>
    <table>
        <tr>
            <th>Employee Name</th>
            <th>Salary</th>
        </tr>
        <tr>
            <td>Jack</td>
            <td>60,000</td>
        </tr>
        <tr>
            <td>Bartowski</td>
            <td>70,000</td>
        </tr>
    </table>
</body>
</html>
```

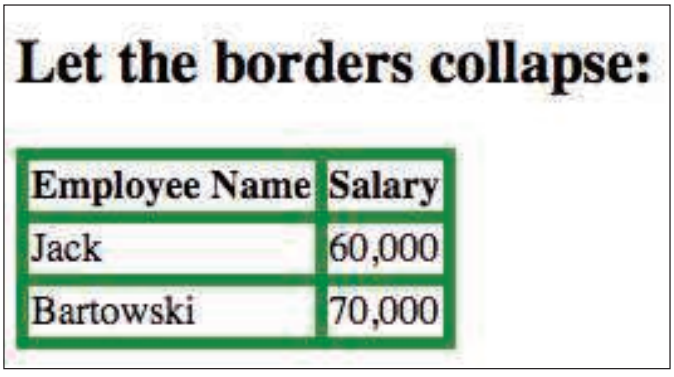
This code shows the following result in a browser window where table and cell borders are in green color.



To collapse borders of a table and merge them in a single border, the *border-collapse* property is used.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    table {
        border-collapse: collapse;
    }

    table, td, th {
        border: 5px solid green;
    }
    </style>
</head>
<body>
    <h2>Let the borders collapse:</h2>
    <table>
        <tr>
            <th>Employee Name</th>
            <th>Salary</th>
        </tr>
        <tr>
            <td>Jack</td>
            <td>60,000</td>
        </tr>
        <tr>
            <td>Bartowski</td>
            <td>70,000</td>
        </tr>
    </table>
</body>
</html>
```

This code shows the following result in a browser window where the table and cell borders are in green color.



To define the height and width of a table, CSS provides the height and width properties. We use these in the example below.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    table, td, th {
        border: 1px solid green;
    }

    table {
        border-collapse: collapse;
        width: 70%;
    }

    th {
        height: 70px;
    }
    </style>
</head>
<body>
    <h2>Example of height and width</h2>
    <table>
        <tr>
            <th>Employee Name</th>
            <th>Salary</th>
        </tr>
        <tr>
            <td>Jack</td>
            <td>60,000</td>
        </tr>
        <tr>
            <td>Bartowski</td>
            <td>70,000</td>
        </tr>
    </table>
</body>
</html>
```

This code shows the following result in a browser window.

Example of height and width

Employee Name	Salary
Jack	60,000
Bartowski	70,000

Likewise, you can use the text-align and vertical-align properties to add horizontal and vertical alignments, respectively.

4.17 | Responsiveness

With so many screen sizes, responsiveness is a necessary requirement. What if your screen is too small? In that case a normal table would be unable to display it contents. To address this issue, you can use the "*overflow-x:auto*" property; it adds responsiveness in the table.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    table {
        border-collapse: collapse;
        width: 100%; }
    th, td {
        text-align: left;
        padding: 8px; }
    tr:nth-child(even) {
        background-color: dodgerblue; }
    </style>
</head>
<body>
    <h2>Responsiveness in the Table</h2>
    <p>When the screen becomes small, this table can resize according and even add a scrollbar
so users can read the complete information.</p>
    <div style="overflow-x: auto;">
        <table>
            <tr>
                <th>Student Name</th>
                <th>Roll Number</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th>
                <th>Marks</th></tr>
            <tr>
                <td>Tim</td>
                <td>01</td>
                <td>50</td>
                <td>60</td>
                <td>70</td>
                <td>80</td>
                <td>90</td>
                <td>60</td>
                <td>70</td>
                <td>80</td>
                <td>90</td>
                <td>60</td></tr>
            <tr>
                <td>Steve</td>
                <td>02</td>
                <td>55</td>
                <td>65</td>
                <td>75</td>
                <td>85</td>
                <td>95</td>
                <td>55</td>
                <td>65</td>
                <td>75</td>
                <td>85</td>
                <td>95</td></tr>
            <tr>
                <td>Cory</td>
                <td>03</td>
                <td>72</td>
                <td>82</td>
                <td>72</td>
                <td>82</td>
                <td>62</td>
                <td>52</td>
                <td>92</td>
                <td>62</td>
                <td>72</td>
                <td>92</td></tr>
        </table>
    </div>
</body></html>
```

This code shows the following result in a browser window.

Responsiveness in the Table

When the screen becomes small, this table can resize according and even add a scrollbar so users can read the complete information.

Student Name	Roll Number	Marks	Marks	Marks	Marks	Marks	Marks	Marks	Marks	Marks	Marks
Tim	01	50	60	70	80	90	60	70	80	90	60
Steve	02	55	65	75	85	95	55	65	75	85	95
Cory	03	72	82	72	82	62	52	92	62	72	92

Notice, we used "*nth-child (even)*" property in the table. This property applies the background color only on the even elements in the table. Similarly, you can also add "*odd*" arrangement in the property.

QUICK CHALLENGE List all different types of devices available in the market which have a browser to view HTML pages.

4.18 | Position Property in CSS

To define the positioning of an HTML element, the position property is used. There are five types of values in this property: *static*, *relative*, *fixed*, *absolute*, and *sticky*. After adding these properties, you can make use of the *left*, *right*, *top*, and *bottom* properties to place your elements on the webpage.

By default, all elements are static. Elements which are static are not disrupted by the *left*, *right*, *top*, and *bottom* properties. Let us see the following example,

```
<!DOCTYPE html>
<html>
<head>
   <style>
   div.s {
      position: static;
      border: 3px solid blue;
   }
   </style>
</head>
<body>
   <h2>Static Example</h2>
   <div class="s">This is an example of static position.</div>
</body>
</html>
```

This code shows the following result in a browser window.

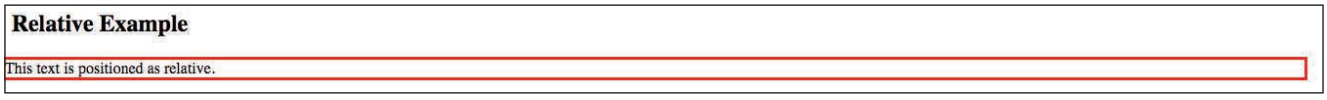
Static Example

This is an example of static position.

When position: relative is used with an element then it is placed relative to its position. However, if you use the *left*, *right*, *top*, and *bottom* properties with it, then the element would be disrupted and displaced from its position.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div.rel {
        position: relative;
        right: 10px;
        border: 3px solid red;
    }
    </style>
</head>
<body>
    <h2>Relative Example</h2>
    <div class="rel">This text is positioned as relative.</div>
</body>
</html>
```

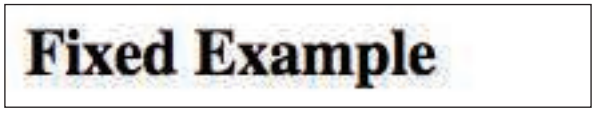
This code shows the following result in a browser window.



If you use the fixed position, then your element is placed relative with respect to the viewport. Therefore, if a visitor scrolls it, the placement of the element would not be changed. For positioning it, the *left*, *right*, *top*, and *bottom* properties are utilized.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div.fx {
        position: fixed;
        bottom: 0;
        right: 0;
        width: 200px;
        border: 5px solid blue;
    }
    </style>
</head>
<body>
    <h2>Fixed Example</h2>
    <div class="fx">This element has a fixed property.</div>
</body>
</html>
```

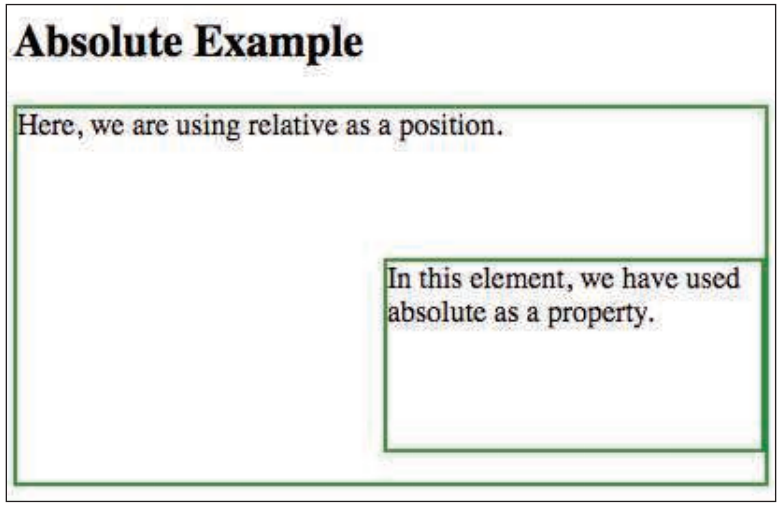
This code shows the following result in a browser window.



An *absolute position* is that in which the element is placed with respect to the closest placed ancestor. In case, it does not have one, it assumes the document's body to be the one.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    div.rel {
        position: relative;
        width: 400px;
        height: 200px;
        border: 2px solid green;
    }

    div.abs {
        position: absolute;
        top: 80px;
        right: 0;
        width: 200px;
        height: 100px;
        border: 2px solid green;
    }
    </style>
</head>
<body>
    <h2>Absolute Example</h2>
    <div class="rel">
        Here, we are using relative as a position.
        <div class="abs">In this element, we have used absolute as a property.</div>
    </div>
</body>
</html>
```

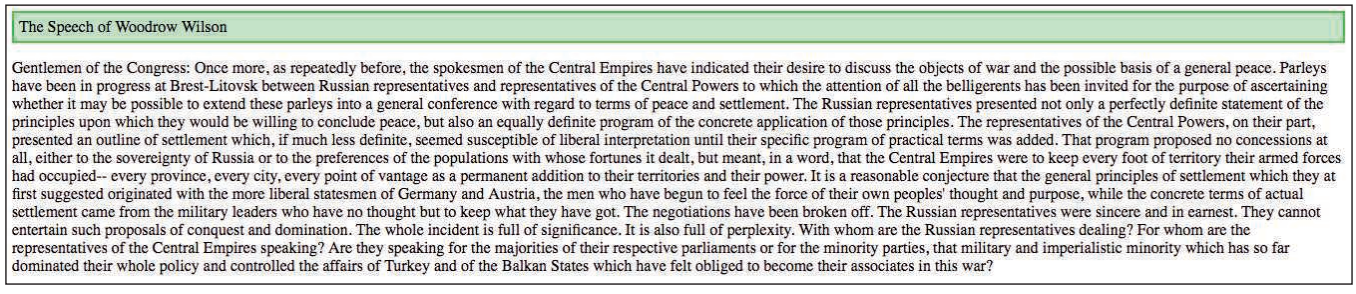
This code shows the following result in a browser window.



An HTML element whose position has been set to "sticky" is placed according to the scroll position of a user. The scroll position can either make it relative or fixed. Let us execute the following code to understand this better.

```
<!DOCTYPE html>
<html>
<head>
   <style>
   div.sp {
      position: sticky;
      top: 0;
      padding: 5px;
      background-color: #cae8ca;
      border: 2px solid #4CAF50;
   }
   </style>
</head>
<body>
   <div class="sp">The Speech of Woodrow Wilson</div>
   <div style="padding-bottom: 2000px">
      <p>Gentlemen of the Congress: Once more, as repeatedly before, the spokesmen of
the Central Empires have indicated their desire to discuss the objects of war and the
possible basis of a general peace. Parleys have been in progress at Brest-Litovsk between
Russian representatives and representatives of the Central Powers to which the attention
of all the belligerents has been invited for the purpose of ascertaining whether it may
be possible to extend these parleys into a general conference with regard to terms of
peace and settlement. The Russian representatives presented not only a perfectly defi-
nite statement of the principles upon which they would be willing to conclude peace, but
also an equally definite program of the concrete application of those principles. The
representatives of the Central Powers, on their part, presented an outline of settlement
which, if much less definite, seemed susceptible of liberal interpretation until their
specific program of practical terms was added. That program proposed no concessions at
all, either to the sovereignty of Russia or to the preferences of the populations with
whose fortunes it dealt, but meant, in a word, that the Central Empires were to keep
every foot of territory their armed forces had occupied every province, every city, every
point of vantage as a permanent addition to their territories and their power. It is a
reasonable conjecture that the general principles of settlement which they at first sug-
gested originated with the more liberal statesmen of Germany and Austria, the men who
have begun to feel the force of their own peoples' thought and purpose, while the con-
crete terms of actual settlement came from the military leaders who have no thought but
to keep what they have got. The negotiations have been broken off. The Russian repre-
sentatives were sincere and in earnest. They cannot entertain such proposals of conquest
and domination. The whole incident is full of significance. It is also full of perplexity.
With whom are the Russian representatives dealing? For whom are the representatives of
the Central Empires speaking? Are they speaking for the majorities of their respective
parliaments or for the minority parties, that military and imperialistic minority which
has so far dominated their whole policy and controlled the affairs of Turkey and of the
Balkan States which have felt obliged to become their associates in this war?</p>
   </div>
</body>
</html>
```

This code shows the following result in a browser window.

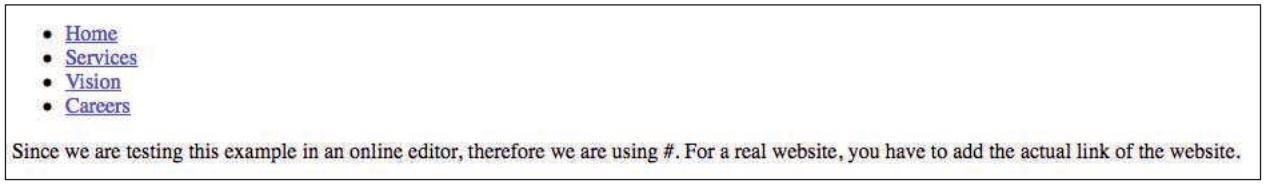


4.19 | Navigation Bars

Navigation bars are one of the most important components of a website. CSS assists web designers in designing aesthetically pleasing navigation bars. A navigation bar is basically a list of links, therefore, it obviously uses HTML as a foundation. Check this basic navigation bar in HTML.

```
<!DOCTYPE html>
<html>
<body>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#vision">Vision</a></li>
        <li><a href="#careers">Careers</a></li>
    </ul>
    <p>Since we are testing this example in an online editor, therefore we are using #.
For a real website, you have to add the actual link of the website.</p>
</body>
</html>
```

This code shows the following result in a browser window.



But we do not use bullet points in the navigation bar. So, remove them. The margin and padding setting is done to adjust the elements to eliminate the default setting of browsers.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
    }
    </style>
</head>
<body>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#vision">Vision</a></li>
        <li><a href="#careers">Careers</a></li>
    </ul>
    <p>Since, we are testing this example in an online editor, therefore we are using #.
For a real website, you have to add the actual link of the website.</p>
</body>
</html>
```

This code shows the following result in a browser window.

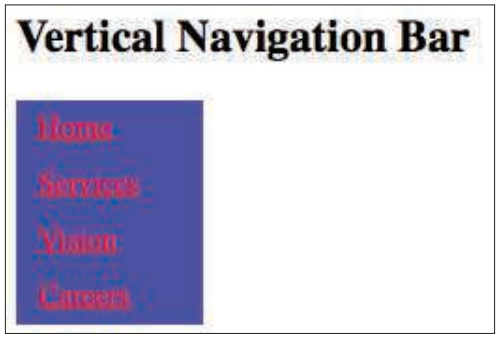
Home
Services
Vision
Careers

Since, we are testing this example in an online editor, therefore we are using #. For a real website, you have to add the actual link of the website.

In the above example, we generated a basic vertical navigation bar. Now, let us construct another one and add some color to its background. Additionally, we have ensured that whenever a user moves a cursor around the menu contents, the color of the bar would change according to our preference.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    ul {
        list-style-type: none;
        padding: 0;
        width: 100px;
        background-color: blue;
    }

    li a {
        display: block;
        color: red;
        padding: 6px 12px;
    }
    /* Updating the color when the cursor hovers around it */
    li a:hover {
        background-color: dodgerblue;
        color: green;
    }
    </style>
</head>
<body>
    <h2>Vertical Navigation Bar</h2>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#vision">Vision</a></li>
        <li><a href="#careers">Careers</a></li>
    </ul>
</body>
</html>
```

This code shows the following result in a browser window where ul element background is in blue.



Similarly, let us build a horizontal navigation bar. To do this, the *<li>* elements are "floated". This is done by using a property "*float*" so the block elements could be slided with each other. Additionally, we have added a class "*active*" in our list to show the current webpage to the user.

```
<!DOCTYPE html>
<html>
<head>
    <style>
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        overflow: hidden;
        background-color: brown;
    }

    li {
        float: left;
    }

    li a {
        display: block;
        color: orange;
        padding: 12px 15px;
    }

    li a:hover {
        background-color: yellow;
    }
    </style>
</head>
<body>
    <ul>
        <li><a class="active" href="#home">Home</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#vision">Vision</a></li>
        <li><a href="#careers">Careers</a></li>
    </ul>
</body>
</html>
```

This code shows the following result in a browser window where ul element is shown in brown and in the second image "Careers" li element is hovered by a user which background is shown in yellow as set for li a:hover property.

Flash Quiz

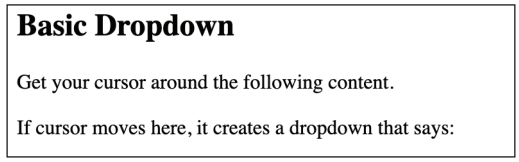
List all the uses of navigation bar styling.

4.20 | Dropdown

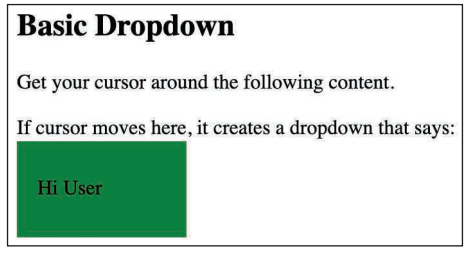
When a user points at an element with a cursor, a dropdown box appears in that area of the webpage. To do this, observe the following example,

```
<!DOCTYPE html>
<html>
<head>
    <style>
    .dropdown {
        position: relative;
        display: inline-block;
    }
    .dropdown-content {
        display: none;
        position: absolute;
        background-color: green;
        min-width: 100px;
        box-shadow: 0px 6px 12px 0px rgba(250, 250, 250, 0.2);
        padding: 12px 16px;
    }
    .dropdown:hover .dropdown-content {
        display: block;
    }
    </style>
</head>
<body>
    <h2>Basic Dropdown</h2>
    <p>Get your cursor around the following content.</p>
    <div class="dropdown">
        <span>If cursor moves here, it creates a dropdown that
            says:</span>
        <div class="dropdown-content">
            <p>Hi User</p>
        </div>
    </div>
</body>
</html>
```

This code shows the following result in a browser window before the cursor moves over it.



Upon moving the cursor over to that line, you will see the following output.

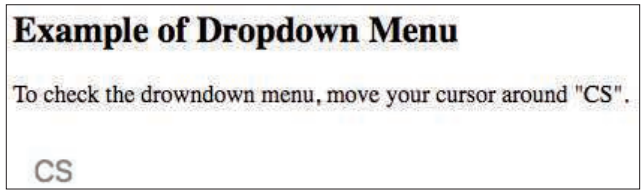


Here we have used a "*dropdown*" class. In order to ensure that the content of the dropdown is mentioned right under the dropdown button, we used a relative position. The *.dropdown-content* class entails any content that is defined for the dropdown. By default, it is hidden on the webpage. To check its content, a user would have to move the cursor around it. The *box-shadow* property is used for giving a card-like look to the dropdown.

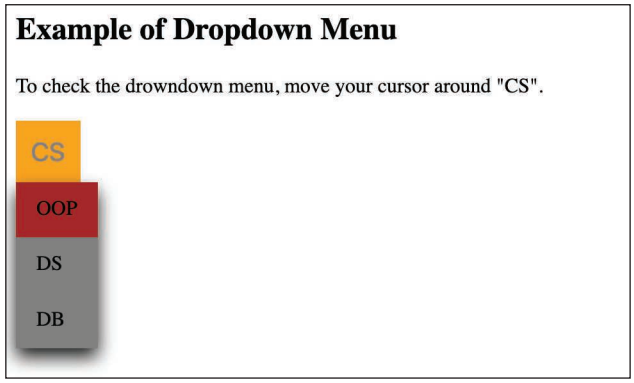
Let us see another example of dropdown menu. In this example, we are showing a list from our dropdown menu. It is identical to the previous example, except our use of links in a list to show options.

```
<!DOCTYPE html>
<html>
<head>
   <style>
   .dropbtn {
      background-color: white;
      color: gray;
      padding: 12px;
      font-size: 20px;
      border: none;
   }

   .dropdown {
      position: relative;
      display: inline-block;
   }

   .dropdown-content {
      display: none;
      position: absolute;
      background-color: gray;
      min-width: 100%;
      box-shadow: 0px 7px 14px 0px rgba(0, 0, 0, 0.9);
   }

   .dropdown-content a {
      color: black;
      padding: 12px 16px;
      text-decoration: none;
      display: block;
   }

   .dropdown-content a:hover {
      background-color: brown
   }

   .dropdown:hover .dropdown-content {
      display: block;
   }

   .dropdown:hover .dropbtn {
      background-color: orange;
   }
   </style>
</head>
<body>
   <h2>Example of Dropdown Menu</h2>
   <p>To check the drowndown menu, move your cursor around "CS".</p>
   <div class="dropdown">
      <button class="dropbtn">CS</button>
      <div class="dropdown-content">
         <a href="#">OOP</a><a href="#">DS</a><a href="#">DB</a>
      </div>
   </div>
</body>
</html>
```

This code shows the following result in a browser window.



Upon moving over the cursor on CS, you will see the following dropdown image.



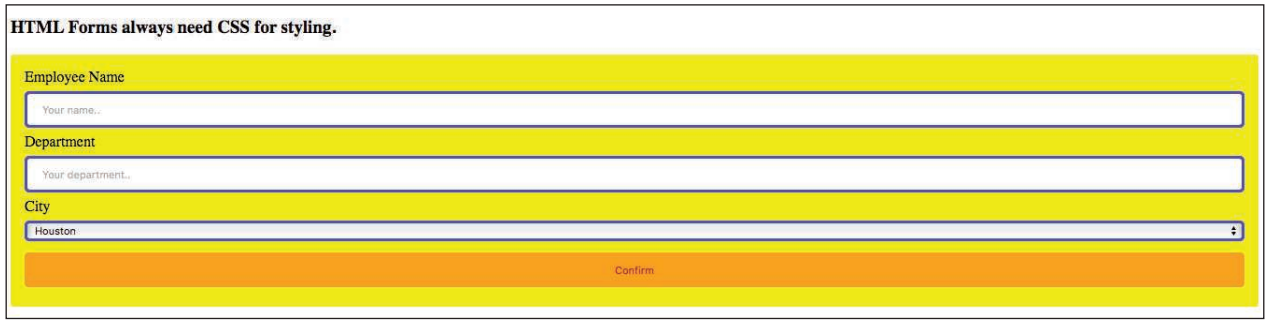
4.21 | Forms

Usually, HTML forms are styled through CSS. The input contents are specified with their types to modify their components in CSS. Consider the following example. Go through our HTML examples and you can easily see the difference in the look of the form.

```
<!DOCTYPE html>
<html>
    <style>
    input[type=text], select {
        width: 100%;
        padding: 10px 16px;
        margin: 6px 0;
        display: inline-block;
        border: 3px solid blue;
        border-radius: 5px;
        box-sizing: border-box;
    }

    input[type=submit] {
        width: 100%;
        background-color: orange;
        color: brown;
        padding: 12px 16px;
        margin: 6px 0;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

    input[type=submit]:hover {
        background-color: green;
    }

    div {
        border-radius: 3px;
        background-color: yellow;
        padding: 15px;
    }
    </style>
```

```
<body>
    <h3>HTML Forms always need CSS for styling.</h3>
    <div>
        <form action="/action_page.php">
            <label for="emp">Employee Name</label><input type="text" id="emp"
                name="employee name" placeholder="Your name..."><label
                for="dept">Department</label><input type="text" id="dept"
                name="department" placeholder="Your department.."><label
                for="city">City</label><select id="city" name="city">
                <option value="Houston">Houston</option>
                <option value="Tampa">Tampa</option>
                <option value="San Francisco">San Francisco</option>
            </select><input type="submit" value="Confirm">
        </form>
    </div>
</body>
</html>
```

This code shows the following result in a browser window where the div element background is in yellow color.



When you hover the cursor over the "Confirm" button, you can see the background color of the button changes to green. The following image shows the button background color changed to green.

Can you add form validations with CSS?

Summary

CSS is a great addition to the front-end technology stack. It not only adds visually appealing design but also carry less weight in terms of page load. CSS is the default method used for web development. It is very simple to learn and use and provides a lot of styling options for all types of HTML elements.

In this chapter, we have learned the following concepts:

1. CSS and its use.
2. Styling various types of HTML elements from a label to table.

3. Id selector to style a specific element.
4. Class selection to style a group of elements.

In Chapter 5, we will learn jQuery and how to use it to add programming logic on HTML pages. jQuery is a JavaScript library, so it offers all the JavaScript features.

Multiple-Choice Questions

1. Name the text property that can be used for aligning a block elements' inline content.
   (a) text-decoration
   (b) text-align
   (c) text-direction
   (d) text-pattern
2. Which of the following properties explains whether an element is an accelerator indicator or not?
   (a) Accelerator
   (b) Jump-start
   (c) Animation
   (d) Push
3. Which one of the following CSS properties is used for padding an element?
   (a) Padding-top
   (b) Padding-bottom
   (c) Padding-left
   (d) All of the above
4. Which one of the following is used to describe elements in CSS?
   (a) Selector
   (b) Value
   (c) Property
   (d) None of the above
5. Which one of the following CSS3 color features can be utilized as a macro for any current color?
   (a) HSL color
   (b) RGB color
   (c) HSLa color
   (d) CurrentColor Keyword

Review Questions

1. What is CSS and how it is useful?
2. What is the use of id selector?
3. What is the use of class selector?
4. How to add external style sheet?
5. Can you use HTML elements as selector to add CSS style?
6. Can you use CSS to make a page look good on all the devices?
7. What happens if multiple style sheets use the same selector?
8. Can you add animation with CSS?

Exercises

1. Create a page to show a HTML table and add CSS to make it look nicer.
2. Create a page to elements. You may add nested elements and add CSS to make it look like a table. See the following example and extend it further.

```
<div>
  <div>Name:</div>
  <div>MayurRamgir</div>
  <div>Address:</div>
  <div>Boston, MA, USA</div>
  <div>Course:</div>
  <div>MS Computational Science & Engi-
neering</div>
</div>
```

3. Create a page with form elements and add CSS to make it look better.
4. Create a vertical menu bar which will stick either on the left or right side of the page.
5. Use the above created menu bar and make it movable so it will stay on the same position even after user scrolls down the page.
6. Add an image as a background to a div element and change the image source on mouse hover. In other words, when user moves a mouse pointer on that image, the image should change.
7. Add a div element and an image element. Write "My Special Div" as inline text in div like My Special Div . Change the text color of div element when user moves mouse over the image and change it back when user moves out of the image.
8. Add a drop down element with multiple values. Style each drop down entry differently like different font, color, size, etc.

Project Idea

Take project idea from the previous chapter where we are creating a dictionary application. Add CSS to make this application look visually appealing.

Recommended Readings

1. Eric Meyer and Estelle Weyl. 2017. *CSS: The Definitive Guide- Visual Presentation for the Web*. Shroff/O'Reilly: New Delhi
2. Thomas Powell. 2017. *HTML & CSS: The Complete Reference*. McGraw Hill Education: New York
3. Laura Lemay, Rafe Colburn, Jennifer Kyrnin. 2016. *Mastering HTML, CSS & Javascript Web Publishing*. BPB Publications: New Delhi
4. The World Wide Web Consortium (W3C) – https://www.w3.org/Style/CSS/Overview.en.html
5. W3School https://www.w3schools.com/css/
6. Mozilla Developer Network (MDN) – https://developer.mozilla.org/en-US/docs/Web/CSS